



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/579,623	05/26/2000	Kemi Y. Ibitayo	Sprint IDF 1415	7837
7590 11/04/2003			EXAMINER	
STEVEN J. FUNK SPRINT LAW DEPARTMENT 6391 Sprint PKWY OVERLAND PARK, KS 66251			MAHMOUDI, HASSAN	
			ART UNIT	PAPER NUMBER
			2175	

DATE MAILED: 11/04/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/579,623

Applicant(s)

IBITAYO ET AL.

Examiner

Tony Mahmoudi

Art Unit

2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 25 July 2003.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-14 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-14 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on \_\_\_\_\_ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

## Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

## Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other:

DOV POPOVICH  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100

## DETAILED ACTION

### *Remarks*

1. In response to communications filed on 25-July-2003, the specification of the disclosure has been amended per applicants' request, deleting the reference made to drawing figure 5.  
Claims 1-14 are presently pending in the application.

### *Claim Rejections - 35 USC § 103*

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hartley et al (U.S. patent No. 6,532,465) in view of McComb et al (U.S. Patent No. 6,006,224.)

As to claim 1, Hartley et al teaches a framework (see Abstract, and see column 7, lines 12-13) for isolating a business component from specific implementations of a datastore (see column 5, lines 12-18, where "isolating a business component from specific implementation of a datastore" is read on "corresponding to one or more business objects that are highly specific to a client's implementation), comprising:

Art Unit: 2175

(a) a database in communication with a business component (see figure 4, where “business component” is illustrated as “business objects 35, and “database” is illustrated as “database tables 39);

(b) a domain object factory in communication with the database (see figure 5, where “database” is illustrated as “external data source”);

(c) a domain object in communication with the domain object factory (see figure 4, where “domain object” is illustrated as “domain objects 37, and see “domain object factory 60 in figure 5); and

(d) a datastore in communication with the domain object (see “domain object factory 60” and “data store 65” in figure 5.)

Hartley et al does not teach a database wrapper.

McComb et al teaches a crucible query system (see Abstract) in which he teaches a database wrapper (see figure 2, and see column 7, lines 51-59.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al to include a database wrapper.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al with the teaching of McComb et al because including a database wrapper would provide the system with “re-useable code that encapsulates procedural code in an application (such as a communication protocol or a database)”, where, “once encapsulated the item becomes an object”, as taught by McComb et al (see column 6, lines 14-17.)

As to claim 2, Hartley et al as modified teaches wherein the database wrapper further comprises a database wrapper interface in communication with the business component (see McComb et al, figure 2, where the “database wrapper 202” is shown interfacing to “database 201” containing “business components”) and a database wrapper implementation implementing the domain object factory (see McComb et al, column 7, lines 55-59, where “implementation” is read on “mapping”, and see Hartley et al, figure s 4 and 5.)

As to claim 3, Hartley et al as modified teaches wherein the domain object factory further comprises a domain object interface in communication with the database wrapper (see Hartley et al, figure 5, and see column 12, lines 6-9) and a domain object factory implementation implementing the domain object (see Hartley et al, figure 5, and see column 11, lines 28-39.)

As to claim 4, Hartley et al as modified teaches wherein the domain object further comprises a domain object interface in communication with the domain object factory (see Hartley et al, figures 4 and 5) and a domain object implementation retrieving data from a datastore (see Hartley et al, column 10, lines 58-67.)

As to claim 5, Hartley et al as modified teaches wherein the domain object interface further comprises a transient data converter for converting the domain object from a persistent state to a transient state (see Hartley et al, figure 13, and see column 16, lines 6-51,

Art Unit: 2175

where “objects from a persistent state” is read on “data formats from outside sources”, and “transient state” is read on “standard formats”, and see column 21, lines 63-65.)

As to claims 6 and 11, Hartley et al as modified teaches wherein the datastore is a relational database (see Hartley et al, column 12, lines 6-12, and see column 21, lines 18-19, where “datastore” is read on “data storage device”).)

As to claims 7 and 12, Hartley et al as modified teaches wherein the datastore is an object database (see Hartley et al, column 12, lines 12-14, where “object database” is read on “the master database including the stored domain objects”).)

As to claims 8 and 13, Hartley et al as modified teaches wherein the datastore is accessed remotely (see Hartley et al, column 8, lines 33-37, and see column 21, lines 30-32.)

As to claim 9, Hartley et al teaches a method (see Abstract) for isolating a business component from specific implementations of a datastore (see column 5, lines 12-18, where “isolating a business component from specific implementation of a datastore” is read on “corresponding to one or more business objects that are highly specific to a client’s implementation), comprising:

Art Unit: 2175

(a) interfacing a database to a business component (see figure 4, where “business component” is illustrated as “business objects 35, and “database” is illustrated as “database tables 39);

(c) interfacing a domain object factory to the database (see figure 5, where “database” is illustrated as “external data source”);

(d) implementing the domain object factory (see figure 5, and see column 11, lines 28-39);

(e) interfacing a domain object to the domain object factory (see figures 4 and 5); and

(f) implementing the domain object to retrieve data from a datastore (see column 10, lines 58-67.)

Hartley et al does not teach a database wrapper; and implementing the database wrapper.

McComb et al teaches a crucible query system (see Abstract) in which he teaches a database wrapper; and implementing a database wrapper (see figure 2, and see column 7, lines 51-59, where “implementing” is read on “mapping”.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al to include a database wrapper; and implementing a database wrapper.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al with the teaching of McComb et al because including and implementing a database wrapper, would provide the system with “re-useable code that encapsulates procedural code in an application (such as a communication

Art Unit: 2175

protocol or a database)", where, "once encapsulated the item becomes an object", as taught by McComb et al (see column 6, lines 14-17.)

As to claim 10, Hartley et al as modified teaches the method further comprising converting data received from the datastore from a persistent state to a transient state (see Hartley et al, figure 13, and see column 16, lines 6-51, where "objects from a persistent state" is read on "data formats from outside sources", and "transient state" is read on "standard formats", and see column 21, lines 63-65.)

As to claim 14, Hartley et al teaches a method for (see Abstract) isolating a business component from specific implementations of a datastore (see column 5, lines 12-18, where "isolating a business component from specific implementation of a datastore" is read on "corresponding to one or more business objects that are highly specific to a client's implementation), comprising:

- (a) supplying a database (see figure 2);
- (c) using the database to obtain a domain object factory (see figure 5);
- (d) using the domain object factory to create a domain object (see figure 5, and see column 11, lines 28-39);
- (e) converting the domain object from a persistent state to a transient state (see figure 13, and see column 16, lines 6-51, where "objects from a persistent state" is read on "data formats from outside sources", and "transient state" is read on "standard formats", and see column 21, lines 63-65);



Art Unit: 2175

Hartley et al does not teach:

supplying a database wrapper;

using the database wrapper to begin a database session;

ending the database session; and

returning the domain object to the business component.

McComb et al teaches a crucible query system (see Abstract) in which he teaches:

supplying a database wrapper (see figure 2);

using the database wrapper (see column 7, lines 51-59) to begin a database session (see column 8, lines 41-42, where “begin a database session” is read on “open a connection to the specified database”);

ending the database session (see column 8, line 43, where “end the database session” is read on “close a database connection”); and

returning the domain object to the business component (see column 9, lines 61-64.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al to include supplying a database wrapper; using the database wrapper to begin a database session; ending the database session; and returning the domain object to the business component.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al with the teachings of McComb et al, because supplying a database wrapper would provide the system with “re-useable code that encapsulates procedural code in an application (such as a communication protocol or a database)”, where, “once encapsulated the item becomes an object”, as taught by McComb et

Art Unit: 2175

al (see column 6, lines 14-17), and because using the database wrapper to begin a database session; ending the database session; and returning the domain object to the business component, would enable the system to communicate with various databases using standard database query languages, and therefore, would increase the flexibility and inter-operability of the system.

4. Claims 5 and 10 are further rejected under 35 U.S.C. 103(a) as being unpatentable over Hartley et al (U.S. patent No. 6,532,465) in view of McComb et al (U.S. Patent No. 6,006,224), as applied to claims 1-14 above, and further in view of Brownell et al (U.S. Patent No. 6,009,266.)

As to claims 5 and 10, in addition to the teaching of Hartley et al, as modified, on converting domain object/data from a persistent state to a transient state (see Hartley et al, figure 13, and see column 16, lines 6-51, where “objects from a persistent state” is read on “data formats from outside sources”, and “transient state” is read on “standard formats”, and see column 21, lines 63-65), Brownell et al teaches methods, apparatus and data structures for managing transient and persistent distributed objects (see Abstract), in which he teaches a transient data converter for converting the domain object from a persistent state to a transient state (see figure 4a, and see column 11, lines 1-17.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al as modified to include a transient data converter for converting the domain object from a persistent state to a transient state.

Art Unit: 2175

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Hartley et al as modified, by the teaching of Brownell et al, because a transient data converter for converting the domain object from a persistent state to a transient state, would enable the system to move object data from permanent storage devices into a temporary storage device, such as cache memory, so that the objects can be in transit and can be shared/distributed to other computers which are a part of the distributed object operating environment.

***Response to Arguments***

5. Applicant's arguments filed on 25-July-2003 with respect to the cited references have been fully considered but they not deemed persuasive:

In response to applicants' arguments that "a prima facie case of obviousness does not exist as there is no teaching or suggestion to combine Hartley and McComb", and that applicants "disagree that McComb provides any motivation or suggestion to combine a database wrapper with the other elements in the manner disclosed and claimed by applicants", and that there is "no teaching or suggestion to further combine Brownell", and further that "this patchwork of individual elements from these disparate references lacking any common thread or teaching is insufficient to teach or suggest applicants' claimed combination as a whole", the arguments have been fully considered but are not deemed

Art Unit: 2175

persuasive, because the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, all cited references teach inventions that are in the same field of endeavor and examiner is establishing motivations in the knowledge generally available to one of ordinary skill in the art, to first modify the invention of Hartley et al with the teaching of McComb et al because including a database wrapper would provide the system with “re-useable code that encapsulates procedural code in an application (such as a communication protocol or a database)”, where, “once encapsulated the item becomes an object”, as taught by McComb et al (see column 6, lines 14-17), and later, to combine Hartley et al as modified, by the teaching of Brownell et al, because a transient data converter for converting the domain object from a persistent state to a transient state, would enable the system to move object data from permanent storage devices into a temporary storage device, such as cache memory, so that the objects can be in transit and can be shared/distributed to other computers which are a part of the distributed object operating environment.

In response to applicants’ arguments that “Hartley does not teach or suggest an additional abstraction layer between the domain object factory and the business component as provided by the database wrapper”, the argument has been fully considered but is not deemed persuasive because “an additional abstraction layer” is not recited in the rejected claim.

Art Unit: 2175

Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Further, “database wrapper” is taught by McComb et al as discussed in the previous office action (see McComb et al, column 7, lines 51-59, and see figure 2.)

In response to applicants’ argument that “McComb does NOT teach the use of a database wrapper to encapsulate a domain object factory”, the argument has been fully considered but is not deemed persuasive, because McComb et al teaches “a ‘wrapper’ comprises re-useable code that ‘encapsulates’ procedural code in an application [such as a communication protocol or a database]. Once ‘encapsulated’ the ‘item’ becomes an ‘object’ (see McComb et al, column 6, lines 14-17.)

In response to applicants’ argument that Brownell et al “does not show the conversion of a domain object”, the argument has been fully considered but is not deemed persuasive, because Brownell et al teaches this “conversion” in “bringing” of objects from persistent state into transient state, as discussed in the previous office action (see figures 4a and 4b, and see column 11, lines 1-39.)

Art Unit: 2175

*Conclusion*

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).


A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

7. Any inquiries concerning this communication or earlier communications from the examiner should be directed to Tony Mahmoudi whose telephone number is (703) 305-4887. The examiner can normally be reached on Mondays-Fridays from 08:00 am to 04:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dov Popovici, can be reached at (703) 305-3830.

tm

September 24, 2003

  
DOV POPOVICI  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100